UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/802,291	03/08/2001	Ashley Saulsbury	16747015310	6894

20350 7590 06/04/2004

TOWNSEND AND TOWNSEND AND CREW, LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

EXAMINER	
O BRIEN, BARRY J	
ART UNIT	PAPER NUMBER
2183	7

DATE MAILED: 06/04/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

4

Office Action Summary	**Application No.** 09/802,291	**Applicant(s)** SAULSBURY, ASHLEY	
	Examiner Barry J. O'Brien	**Art Unit** 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 April 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-6 and 8-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-6 and 8-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-6 and 8-20 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed on record in the file: Amendment A as received on 4/7/04.

Specification

3. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

4. The applicant is requested to review the specification and update the status of all co-pending applications made mention of, replacing attorney docket numbers with current U.S. application or patent numbers when appropriate.

Improper Amendment

5. Applicant's cancellation of claim 7 in the present amendment, filed 4/7/04, has failed to adhere to 37 CFR 1.121. As is clearly seen from the attached office flyer, "canceled and not entered claims must be identified by only the claim number and status". Applicant's amendment fails to identify canceled claim 7 by **only the claim number and status**. Applicant is advised to carefully review the requirements of 37 CFR 1.121 regarding amendment formatting as

summarized in the attached flyer, and to assure that all future amendments properly adhere to these requirements.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1, 3, 5-6, 8-9, 11-12, 14, 16-17 and 20 are rejected under 35 U.S.C. 102(b) as being anticipated by the *Alpha Architecture Handbook* (hereinafter "Alpha").

8. Regarding claim 1, Alpha has taught a processing core that executes a compare instruction (see p.4-113 Sec. 4.10.8, CMPTLE instruction), the processing core comprising:

a. A plurality of general-purpose registers comprising a first input operand register, a second input operand register and an output operand register (see p.3-2 Sec. 3.1.3),

b. A register file comprising the plurality of general-purpose registers (see p.3-2 Sec. 3.1.3),

c. Comparison logic coupled to the register file, wherein the comparison logic tests for at least two of the following relationships with the compare instruction alone: less than, equal to, greater than and no valid relationship (see p.4-113 Sec. 4.10.8, CMPTLE instruction). Here, the CMPTLE instruction performs both the less than and equal comparisons using one instruction. Also, because less than or equal is a mutually exclusive operation with greater than, the CMPTLE instruction inherently determines if an operand is greater than a second operand by determining if its not less than or equal to the second operand.

d. Decode logic which selects the output operand register from the plurality of general-purpose registers (see p.4-113 Sec. 4.10.8). Here, because the IEEE Floating Compare instruction includes a target register (Fc) which stores the outputted results of a comparison, it is inherent that decoding logic be present in any processor which implements the Alpha instruction set architecture so that the specified target register can be correctly located and written into.

e. A store path between the comparison logic and the selected output operand register (see p.4-113 Sec. 4.10.8). Here, because the IEEE Floating Compare instruction includes a target register (Fc) which stores the outputted results of a comparison, it is inherent that a path exist between the logic which performs the comparison and the target register specified by the instruction so that the specified target register can be correctly located and written into.

9. Regarding claim 3, Alpha has taught the processing core that executes the compare instruction as set forth in claim 1, wherein said decode logic selects the first and second input operand registers from the plurality of general-purpose registers (see p.4-113 Sec. 4.10.8). Here, because the IEEE Floating Compare instruction includes two operand registers (Fa and Fb), it is inherent that decoding logic be present in any processor which implements the Alpha instruction set architecture so that the specified operand registers can be correctly located and read from.

10. Regarding claim 5, Alpha has taught the processing core that executes the compare instruction as set forth in claim 1, further comprising:

a. A first load path between the first input operand register and the comparison logic (see p.4-113 Sec. 4.10.8),

b. A second load path between the second input operand register and the comparison logic (see p.4-113 Sec. 4.10.8). Here, because the IEEE Floating Compare instruction includes two operand registers (Fa and Fb) which contain the operands for the comparison, it is inherent that a path exist between the logic which performs the comparison and each of the operand registers specified by the instruction so that the specified operands can be correctly located and read from, allowing the comparison to be made correctly.

11. Regarding claim 6, Alpha has taught the processing core that executes the compare instruction as set forth in claim 1, wherein the output operator register stores a value indicating a relationship between the first and second input operator registers which is at least one of greater than, less than, equal to and not a number (see p.4-113, Sec. 4.10.8, Description).

12. Regarding claim 8, Alpha has taught the processing core that executes the compare instruction as set forth in claim 6, wherein the value is an integer (see p.4-113, Sec. 4.10.8, Description). Here, when the comparison result is false, a value of zero is written into the target register, with zero being defined as an integer.

13. Regarding claim 9, Alpha has taught the processing core that executes the compare instruction as set forth in claim 1, wherein:

a. The first input operand register is a double precision floating point data type (see p.3-2 Sec. 3.1.3, p.4-62 Sec. 4.7, and p.4-113 Sec.4.10.8),

b. The second input operand register is a single precision floating point data type (see p.3-2 Sec. 3.1.3, p.4-62 Sec. 4.7, and p.4-113 Sec.4.10.8),

c. The output operand register is a double precision floating point data type (see p.3-2 Sec. 3.1.3, p.4-62 Sec. 4.7, and p.4-113 Sec.4.10.8).

14. Here, Alpha has taught the IEEE Floating Compare instruction using the T_Floating data format (see p.4-113 Sec.4.10.8), which is defined as the IEEE Double Precision floating point standard (see p.4-62 Sec 4.7). If one of the registers in the instruction, whether it is an input operand register or the output operand register, is single precision, the data is stored as a single precision value (see p.4-62 Sec. 4.7) but operated on correctly by instructions that use double precision values, such as the Floating Compare instruction (see p.3-2 Sec. 3.1.3).

15. Regarding claim 11, Alpha has taught the processing core that executes the compare instruction as set forth in claim 1, wherein the register file comprises special purpose registers which cannot store an output operand (see p.3-2 Sec. 3.1.3). Here, the registers F31 is a special purpose register in the floating-point register file which only supplies and stores the value of zero regardless of what is written to it.

16. Regarding claim 12, Alpha has taught a method for performing a compare operation, the method comprising the steps of:

a. Decoding a compare instruction (see p.4-113 Sec. 4.10.8). Here, because the IEEE Floating Compare instruction includes two operand registers (Fa and Fb) and a target register (Fc), it is inherent that decoding logic be present in any processor which implements the Alpha instruction set architecture so that the operand registers can be located and read from and the specified target register can be correctly located and written into.

b. Configuring first and second paths between a register file and comparison logic (see p.4-113 Sec. 4.10.8). Here, because the IEEE Floating Compare instruction includes two operand registers (Fa and Fb) which contain the operands for the comparison, it is inherent that a path exist between the logic which performs the comparison and each of the operand registers specified by the instruction so that the specified operands can be correctly located and read from, allowing the comparison to be made correctly.

c. Configuring a third path between the comparison logic and the register file (see p.4-113 Sec. 4.10.8). Here, because the IEEE Floating Compare instruction includes a target register (Fc) which stores the outputted results of a comparison, it is inherent that a path exist between the logic which performs the comparison and the target register specified by the instruction so that the specified target register can be correctly located and written into.

d. Comparing a first input operand and a second input operand with the compare operation alone to produce a result which indicates at least two of the following mathematical relationships between the first input operand and the second input operand in the alternative: less than, equal to, greater than and no valid relationship (see p.4-113 Sec. 4.10.8, CMPTLE instruction). Here, the CMPTLE instruction performs both the less than and equal comparisons using one instruction. Also, because less than or equal is a mutually exclusive operation with greater than, the CMPTLE instruction inherently determines if an operand is greater than a second operand by determining if its not less than or equal to the second operand.

e. Coupling an output operand to a general-purpose register in the register file (see p.4-113 Sec. 4.10.8). Here, because the IEEE Floating Compare instruction includes a target register (Fc) which stores the outputted results of a comparison, it is inherent that a path exist between the logic which performs the comparison and the target register specified by the instruction so that the specified target register can be correctly located and written into.

17. Regarding claim 14, Alpha has taught the method for performing the compare operation as set forth in claim 12, wherein the configuring steps each comprise a step of addressing a general-purpose register in the register file (see p.4-113 Sec. 4.10.8). Here, because the IEEE Floating Compare instruction includes two operand registers (Fa and Fb) and a target register (Fc), it is inherent that decoding logic be present in any processor which implements the Alpha instruction set architecture so that the operand registers can be located and read from and the specified target register can be correctly located and written into. Furthermore, because a register can't be read from or written to without knowing its address, inherent to the locating of the operand registers contained within the IEEE Floating Compare instruction is the addressing of those registers within the register file so they can be read out and/or written to.

18. Regarding claim 16, Alpha has taught the method for performing the compare operation as set forth in claim 12, wherein the comparing step comprises a step of converting a data type of

at least one of the first and second input operands (see p.4-62 Sec. 4.7). Here, it is taught that data conversion takes place between single- and double-precision floating-point instructions. Alpha has taught the IEEE Floating Compare instruction using the T_Floating data format (see p.4-113 Sec.4.10.8), which is defined as the IEEE Double Precision floating point standard (see p.4-62 Sec 4.7). If one of the registers in the instruction, whether it is an input operand register or the output operand register, is single precision, the data is stored as a single precision value (see p.4-62 Sec. 4.7) but operated on correctly by instructions that use double precision values, such as the Floating Compare instruction (see p.3-2 Sec. 3.1.3).

19. Regarding claim 17, Alpha has taught a method for executing a compare instruction in a processor, the method comprising steps of:

a. Issuing the compare operation. While not taught explicitly, it is inherent in the operation of a processor which implements an instruction set that in order to execute an instruction it must be issued to the execution unit(s).

b. Comparing a first input operand and a second input operand to determine at least two mathematical relationships between the first and second input operands, wherein the compare instruction alone causes the comparing step (see p.4-113 Sec. 4.10.8, CMPTLE instruction). Here, the CMPTLE instruction performs both the less than and equal comparisons using one instruction. Also, because less than or equal is a mutually exclusive operation with greater than, the CMPTLE instruction inherently determines if an operand is greater than a second operand by determining if its not less than or equal to the second operand.

c. Determining an output operand indicative of the mathematical relationships (see p.4-113 Sec. 4.10.8, Description),

d. Storing the output operand in a general-purpose register of a register file (see p.4-113 Sec. 4.10.8, Description).

20. Regarding claim 20, Alpha has taught the method for executing the compare instruction in the processor as set forth in claim 17, wherein the general-purpose register is used to store operators from other types of instructions (see p.3-2 Sec.3.1.3). Here, the floating-point registers are general-purpose registers that are used as sources and targets for floating-point instructions, which means that the registers are inherently able to store operators from other types of instructions than the IEEE Floating Compare instruction.

Claim Rejections - 35 USC § 103

21. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

22. Claims 2, 4, 10, 15 and 18-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Alpha Architecture Handbook (hereinafter "Alpha"), as applied to claim 1 above, and further in view of Colwell et al., U.S. Patent No. 4,833,599.

23. Regarding claim 2, Alpha has taught the processing core that executes the compare instruction as set forth in claim 1, but has not explicitly taught wherein a very long instruction word includes a plurality of compare instructions.

24. However, Colwell has taught a processor that executes multiple conditional branch instructions that reside in a VLIW instruction in parallel to improve execution speed and reduce the delay associated with branch mis-predictions (see Col.1 line 40 – Col.2 line 18 and Col.3 lines 3-34). One of ordinary skill in the art would have recognized that it is desirable and a goal of microprocessor design to improve the speed and throughput of a microprocessor. Therefore, one of ordinary skill in the art would have found it obvious to modify the processor of the Alpha to process a plurality of conditional branch instructions in a VLIW instruction in parallel to improve the execution speed of the processor.

25. Regarding claim 4, Alpha has taught the processing core that executes the compare instruction as set forth in claim 1, but has not explicitly taught wherein the processing core issues a plurality of compare instructions at one time.

26. However, Colwell has taught a processor that executes multiple conditional branch instructions that reside in a VLIW instruction in parallel to improve execution speed and reduce the delay associated with branch mis-predictions (see Col.1 line 40 – Col.2 line 18 and Col.3 lines 3-34). It is inherent to the parallel execution of multiple instructions then is the parallel issue of multiple instructions. One of ordinary skill in the art would have recognized that it is desirable and a goal of microprocessor design to improve the speed and throughput of a microprocessor, and one method of doing so is to increase the instruction-level parallelism by issuing and subsequently executing multiple instructions in parallel (see Col.1 lines 12-26). Therefore, one of ordinary skill in the art would have found it obvious to modify the processor of the Alpha to issue a plurality of compare instructions at one time so they can be executed in parallel and thus increase the throughput and execution speed of the processor.

27. Regarding claim 10, Alpha has taught the processing core that executes the compare instruction as set forth in claim 1, but has not explicitly taught wherein the processing core further comprises a plurality of processing paths that are coupled to the register file.

28. However, Colwell has taught a processor with multiple processor clusters, each containing multiple processing paths (integer and floating point processors) with each processing path connecting to a register file (see Col.5 lines 47-55), so that multiple instructions can be executed in parallel each cycle, providing an increase in processor performance (see Col.1 lines 7-26). One of ordinary skill in the art would have recognized that it is desirable and a goal of microprocessor design to improve the speed and throughput of a microprocessor. Therefore, one of ordinary skill in the art would have found it obvious to modify the processor of the Alpha to process a plurality of conditional branch instructions in a VLIW instruction in parallel to improve the execution speed of the processor.

29. Regarding claim 15, Alpha has taught the method for performing the compare operation as set forth in claim 12, but has not explicitly taught wherein a compare operation is comprised within a very long instruction word.

30. However, Colwell has taught a processor that executes multiple conditional branch instructions that reside in a VLIW instruction in parallel to improve execution speed and reduce the delay associated with branch mis-predictions (see Col.1 line 40 – Col.2 line 18 and Col.3 lines 3-34). One of ordinary skill in the art would have recognized that it is desirable and a goal of microprocessor design to improve the speed and throughput of a microprocessor. Therefore, one of ordinary skill in the art would have found it obvious to modify the processor of the Alpha

to process a plurality of conditional branch instructions in a VLIW instruction in parallel to improve the execution speed of the processor.

31. Regarding claim 18, Alpha in view of Colwell has taught the method for executing the compare instruction in the processor as set forth in claim 17, but has not explicitly taught wherein the comparing step comprises:

a. Determining if the first input operand is less than the second input operand (see "CMPTLE" instruction of p.4-113 Sec. 4.10.8),

b. Determining if the first input operand is greater than the second input operand (see "CMPTLE" instruction of p.4-113 Sec. 4.10.8). Here, because less than or equal is a mutually exclusive operation with greater than, the CMPTLE instruction inherently determines if an operand is greater than a second operand by determining if its not less than or equal to the second operand.

c. Determining if the first input operand is equal to the second input operand (see "CMPTLE" instruction of p.4-113 Sec. 4.10.8),

d. Determining if there is no valid relationship between the first input operand the second input operand (see "CMPTUN" instruction of p.4-113 Sec. 4.10.8).

32. The *Alpha Architecture Handbook* has taught the above comparisons as parts of two different instructions (see p.4-113 Sec. 4.10.8), but has not explicitly taught a single instruction with the ability to perform the four comparisons simultaneously. However, the comparison logic required to perform the four comparisons as described inherently exists (see paragraph 19 above). One of ordinary skill in the art would have recognized that executing the comparisons of two instructions as a single instruction in a single clock cycle increases the speed and throughput

of a microprocessor. Therefore, one of ordinary skill in the art would have found it obvious to perform the four comparisons simultaneously when processing one instruction instead of separately for each of four instructions in order to increase the speed and throughput of the processor.

33. Regarding claim 19, Alpha has taught the method for executing the compare instruction in the processor as set forth in claim 17, but has not explicitly taught wherein the compare instruction is a very long instruction word which comprises a plurality of compare instructions which are processed in parallel down separate processing paths.

34. However, Colwell has taught a processor that executes multiple conditional branch instructions that reside in a VLIW instruction in parallel to improve execution speed and reduce the delay associated with branch mis-predictions (see Col.1 line 40 – Col.2 line 18 and Col.3 lines 3-34). One of ordinary skill in the art would have recognized that it is desirable and a goal of microprocessor design to improve the speed and throughput of a microprocessor. Therefore, one of ordinary skill in the art would have found it obvious to modify the processor of the Alpha to process a plurality of conditional branch instructions in a VLIW instruction in parallel to improve the execution speed of the processor.

35. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over the *Alpha Architecture Handbook* (hereinafter "Alpha") as applied to claim 12 above, and further in view of Patterson and Hennessy, *Computer Organization and Design* (hereinafter "Patterson").

36. Regarding claim 13, Alpha has taught the method for performing a compare operation as set forth in claim 12, but has not explicitly taught wherein the method further comprises a step of enabling the comparison logic in an arithmetic logic unit.

37. However, Patterson have taught that logical operations, such as comparisons, are conventionally performed within an arithmetic logic unit because they contain the necessary hardware building blocks to perform comparisons, such as AND gates for equality comparisons (see p.230-231). One of ordinary skill in the art would have recognized the desire of a microprocessor designer to reuse hardware in order to minimize the space taken up by dedicated hardware on the chip. Therefore, one of ordinary skill in the art would have found it obvious to modify the processor of the Alpha to include its comparison logic in an arithmetic logic unit, so as to adhere to convention and reuse hardware which already exists, thus minimizing the spaced needed for dedicated hardware on the chip.

Response to Arguments

38. Applicant's arguments filed on 4/7/04 have been fully considered but they are not persuasive.

39. On page 10 of the present amendment, the Applicant argues, in essence:

> "*The amended claims require that two mathematical relationships be determined in a single compare instruction. In stark contrast, the Handbook has four instructions to test for four mathematical relationships, where each instruction tests for a single mathematical relationship. The claimed invention can test for these relationships with a fraction of the instructions to be at least twice as efficient on an instruction-by-instruction basis.*"

40. However, the *Alpha Architecture Handbook* has taught a single compare instruction (the "CMPTLE" instruction) which determines at least two mathematical relationships, namely the

less than, the equal, and the greater than relationships (see paragraph 12 of the previous Office Action dated 1/29/04, and above paragraph 8). Therefore, the prior art of record satisfies the limitations of the invention as claimed.

Conclusion

41. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

42. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

43. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Barry J. O'Brien whose telephone number is (703) 305-5864. The examiner can normally be reached on Mon.-Fri. 6:30am-4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

44. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Barry J. O'Brien
Examiner
Art Unit 2183

BJO
5/20/2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

REVISED AMENDMENT PRACTICE: 37 CFR 1.121 CHANGED
COMPLIANCE IS MANDATORY - Effective Date: July 30, 2003

All amendments filed on or after the effective date noted above must comply with revised 37 CFR 1.121. See Final Rule: **Changes To Implement Electronic Maintenance of Official Patent Application Records** (68 *Fed. Reg.* 38611 (June 30, 2003), posted on the Office's website at: http://www.uspto.gov/web/patents/ifw/ with related information. The amendment practice set forth in revised 37 CFR 1.121, and described below, replaces the voluntary revised amendment format available to applicants since February 2003. **NOTE: STRICT COMPLIANCE WITH THE REVISED 37 CFR 1.121 IS REQUIRED AS OF THE EFFECTIVE DATE (July 30, 2003).** The Office will notify applicants of amendments that are not accepted because they do not comply with revised 37 CFR 1.121 via a Notice of Non-Compliant Amendment. See MPEP 714.03 (Rev. 1, Feb. 2003). The non-compliant section(s) will have to be corrected and the entire corrected section(s) resubmitted within a set period.

Bold underlined italic font has been used below to highlight the major differences between the revised 37 CFR 1.121 and the voluntary revised amendment format that applicants could use since February, 2003.
Note: The amendment practice for **reissues** and **reexamination proceedings**, except for drawings, has not changed.

REVISED AMENDMENT PRACTICE

I. Begin each section of an amendment document on a separate sheet:

Each section of an amendment document (*e.g.*, Specification Amendments, Claim Amendments, Drawing Amendments, and Remarks) must begin on a separate sheet. Starting each separate section on a new page will facilitate the process of separately indexing and scanning each section of an amendment document for placement in an image file wrapper.

II. Two versions of amended part(s) no longer required:

37 CFR 1.121 has been revised to no longer require two versions (a clean version and a marked up version) of each replacement paragraph or section, or amended claim. Note, however, the requirements for a clean version and a marked up version for substitute specifications under 37 CFR 1.125 have been retained.

A) Amendments to the claims:

Each amendment document that includes a change to an existing claim, cancellation of a claim or submission of a new claim, **must include a complete listing** of all claims in the application. After each claim number in the listing, the status must be indicated in a parenthetical expression, and **the text of each pending claim** (with markings to show current changes) must be presented. The claims in the listing will replace all prior claims in the application.

(1) The current status of all of the claims in the application, including any previously canceled, not entered or withdrawn claims, must be given in a parenthetical expression following the claim number using only one of the following seven status identifiers: (original), (currently amended), (canceled), (withdrawn), (new), ***(previously presented) and (not entered).*** The text of all pending claims, ***including withdrawn claims,*** must be submitted each time any claim is amended. Canceled ***and not entered*** claims must be indicated by only the claim number and status, without presenting the text of the claims.

(2) The text of all claims being currently amended must be presented in the claim listing with markings to indicate the changes that have been made relative to the immediate prior version. The changes in any amended claim must be shown by underlining (for added matter) or strikethrough (for deleted matter) with 2 exceptions: (1) for ***deletion of five characters or fewer, double brackets may be used (e.g., [[eroor]]); and (2) if strikethrough cannot be easily perceived (e.g., deletion of the number "4" or certain punctuation marks), double brackets must be used (e.g., [[4]]). As an alternative to using double brackets, however, extra portions of text may be included before and after text being deleted, all in strikethrough, followed by including and underlining the extra text with the desired change*** (*e.g.*, ~~*number 4 as*~~ *number 14 as*). An accompanying clean version is not required and should not be presented. Only claims of the status "currently amended," and "withdrawn" that are being amended, may include markings.

(3) The text of pending claims not being currently amended, ***including withdrawn claims,*** must be presented in the claim listing in clean version, *i.e.*, without any markings. Any claim text presented in clean version will constitute an assertion that it has not been changed relative to the immediate prior version except to omit markings that may have been present in the immediate prior version of the claims.

(4) A claim being canceled must be listed in the claim listing with the status identifier "canceled"; the text of the claim must not be presented. Providing an instruction to cancel is optional.

(5) Any claims added by amendment must be presented in the claim listing with the status identifier "(new)"; the text of the claim must not be underlined.

(6) All of the claims in the claim listing must be presented in ascending numerical order. Consecutive canceled, or not entered, claims may be aggregated into one statement (*e.g.*, Claims 1 – 5 (canceled)).

Example of listing of claims (use of the word "claim" before the claim number is optional):

Claims 1-5 (canceled)

Claim 6 (previously presented): A bucket with a handle.

Claim 7 (withdrawn): A handle comprising an elongated wire.

Claim 8 (withdrawn): The handle of claim 7 further comprising a plastic grip.

Claim 9 (currently amended): A bucket with a ~~green~~ blue handle.

Claim 10 (original): The bucket of claim 9 wherein the handle is made of wood.

Claim 11 (canceled)

Claim 12 (not entered)

Claim 13 (new): A bucket with plastic sides and bottom.

B) Amendments to the specification:

Amendments to the specification, including the abstract, must be made by presenting a replacement paragraph or section or abstract marked up to show changes made relative to the immediate prior version. An accompanying clean version is not required and should not be presented. Newly added paragraphs or sections, including a new abstract (instead of a replacement abstract), must not be underlined. A replacement or new abstract must be submitted on a separate sheet, 37 CFR 1.72. If a substitute specification is being submitted to incorporate extensive amendments, both a clean version (which will be entered) and a marked up version must be submitted as per 37 CFR 1.125.

The changes in any replacement paragraph or section, or substitute specification must be shown by underlining (for added matter) or strikethrough (for deleted matter) with 2 exceptions: (1) for ***deletion of five characters or fewer, double brackets may be used (e.g., [[eroor]]); and (2) if strikethrough cannot be easily perceived (e.g., deletion of the number "4" or certain punctuation marks), double brackets must be used (e.g., [[4]]). As an alternative to using double brackets, however, extra portions of text may be included before and after text being deleted, all in strikethrough, followed by including and underlining the extra text with the desired change*** *(e.g., ~~number 4 as~~ number 14 as)*

C) Amendments to drawing figures:

Drawing changes must be made by presenting replacement figures which incorporate the desired changes and which comply with 37 CFR 1.84. An explanation of the changes made must be presented either in the drawing amendments, or remarks, section of the amendment, ***and may be accompanied by a marked-up copy of one or more of the figures being amended, with annotations.*** Any replacement drawing sheet must ***be identified in the top margin as "Replacement Sheet"*** and include all of the figures appearing on the immediate prior version of the sheet, even though only one figure may be amended. ***Any marked-up (annotated) copy showing changes must be labeled "Annotated Marked-up Drawings" and accompany the replacement sheet in the amendment (e.g., as an appendix)***. The figure or figure number of the amended drawing(s) must **not** be labeled as "amended." If the changes to the drawing figure(s) are not accepted by the examiner, applicant will be notified of any required corrective action in the next Office action. No further drawing submission will be required, unless applicant is notified.

Questions regarding the submission of amendments pursuant to the revised practice set forth in this flyer should be directed to: Elizabeth Dougherty or Gena Jones, Legal Advisors, or Joe Narcavage, Senior Special Projects Examiner, Office of Patent Legal Administration, by e-mail to patentpractice@uspto.gov or by phone at (703) 305-1616.